Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	2810	(719/315.ccls. 719/316.ccls. 717/106.ccls. 717/108.ccls. 717/114.ccls 717/114.ccls. 719/318. ccls.)	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	OFF	2005/12/20 14:10
L2	0	L1 AND (template near2 parameter near2 parameterize near3 correlation)	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	OFF	2005/12/20 14:14
L3	0	L1 AND (template near4 parameter near4 parameterize near4 correlation)	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	OFF	2005/12/20 14:14
L4	0	(template near4 parameter near4 parameterize near4 correlation)	US-PGPUB	OR	OFF	2005/12/20 14:17
L5	323	719/318.ccls.	US-PGPUB	OR	OFF	2005/12/20 15:16
L6	13	template with class with (input or source) with (output or target)	US-PGPUB	OR	OFF	2005/12/20 15:16
L7	23	template with class with (input or source) with (output or target)	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	OFF	2005/12/20 15:17
L8	23	template with class with (input or source) with (output or target)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT	OR	OFF	2005/12/20 15:17
L9	3	nested adj class same template	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT	OR	OFF	2005/12/20 15:19
L10	6	template adj base adj class	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT	OR	OFF	2005/12/20 15:19
L11	108	template with class with parameter$	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT	OR	OFF	2005/12/20 15:20
L12	52	template with (input source) with (output target) with combin$5	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT	OR	OFF	2005/12/20 15:20

Terms used **template** and **inner** **class** Found **348** of **167,655**

Sort results by: relevance
Display results: condensed form
Save results to a Binder
Search Tips
Open results in a new window
Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 Mixin layers: an object-oriented implementation technique for refinements and collaboration-based designs
Yannis Smaragdakis, Don Batory
April 2002 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 11 Issue 2
Publisher: ACM Press
Full text available: pdf(510.43 KB) Additional Information: full citation, abstract, references, citings, index terms

2 Session 2 (short papers): system and practical issues: Collaboration-based evolvable software implementations: Java and Hyper/J vs. C++-templates composition
Nguyen Truong Thang, Takuya Katayama
May 2002 **Proceedings of the International Workshop on Principles of Software Evolution**
Publisher: ACM Press
Full text available: pdf(424.71 KB) Additional Information: full citation, abstract, references

3 Query optimization II: An architecture for query optimization
Arnon Rosenthal, David Reiner
June 1982 **Proceedings of the 1982 ACM SIGMOD international conference on Management of data**
Publisher: ACM Press
Full text available: pdf(1.11 MB) Additional Information: full citation, abstract, references, citings

4 Coordination models, languages and applications (CM): LighTS: a lightweight, customizable tuple space supporting context-aware applications
Gian Pietro Picco, Davide Balzarotti, Paolo Costa
March 2005 **Proceedings of the 2005 ACM symposium on Applied computing**
Publisher: ACM Press
Full text available: pdf(200.86 KB) Additional Information: full citation, abstract, references

5 On the role of language constructs for framework design
Görel Hedin, Jørgen Lindskov Knudsen
March 2000 **ACM Computing Surveys (CSUR)**
Publisher: ACM Press
Full text available: pdf(37.97 KB) Additional Information: full citation, references, index terms

6 Extending Java for high-level Web service construction
Aske Simon Christensen, Anders Møller, Michael I. Schwartzbach
November 2003 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 25 Issue 6
Publisher: ACM Press
Full text available: pdf(947.02 KB) Additional Information: full citation, abstract, references, citings, index terms

7 Efficient crosstalk noise modeling using aggressor and tree reductions
Li Ding, David Blaauw, Pinaki Mazumder
November 2002 **Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design**
Publisher: ACM Press
Full text available: pdf(139.91 KB) Additional Information: full citation, abstract, references, index terms

8 Using testing and JUnit across the curriculum
Michael Wick, Daniel Stevenson, Paul Wagner
February 2005 **ACM SIGCSE Bulletin , Proceedings of the 36th SIGCSE technical symposium on Computer science education SIGCSE '05**, Volume 37 Issue 1
Publisher: ACM Press
Full text available: pdf(219.93 KB) Additional Information: full citation, abstract, references, index terms

9 Change management: An infrastructure for development of object-oriented, multi-level configuration management services
Tien N. Nguyen, Ethan V. Munson, John T. Boyland, Cheng Thao
May 2005 **Proceedings of the 27th international conference on Software engineering**
Publisher: ACM Press
Full text available: pdf(418.74 KB) Additional Information: full citation, abstract, references, index terms

10 Efficient and automatic implementation of the adjoint state method
Mark S. Gockenbach, Daniel R. Reynolds, Peng Shen, William W. Symes
March 2002 **ACM Transactions on Mathematical Software (TOMS)**, Volume 28 Issue 1
Publisher: ACM Press
Full text available: pdf(587.71 KB) Additional Information: full citation, abstract, references, index terms

11 Cluster: an informal report
Shang Lujun
January 1991 **ACM SIGPLAN Notices**, Volume 26 Issue 1
Publisher: ACM Press
Full text available: pdf(1.15 MB) Additional Information: full citation, abstract, index terms

12 Facilitating abstraction and reuse with ExpectTK
Navid Sabbaghi
February 1996 **Crossroads**, Volume 2 Issue 3
Publisher: ACM Press
Full text available: html(36.36 KB) Additional Information: full citation, index terms

13 Dependency diagrams
Mark Ray
February 1996 **Crossroads**, Volume 2 Issue 3
Publisher: ACM Press
Full text available: html(36.36 KB) Additional Information: full citation, index terms

14 An overview of the standard template library

G. Bowden Wise
April 1996 **ACM SIGPLAN Notices**, Volume 31 Issue 4
Publisher: ACM Press
Full text available: pdf(566.70 KB) Additional Information: full citation, index terms

15 Automatic information extraction from large websites
Valter Crescenzi, Giansalvatore Mecca
September 2004 **Journal of the ACM (JACM)**, Volume 51 Issue 5
Publisher: ACM Press
Full text available: pdf(1.13 MB) Additional Information: full citation, abstract, references, index terms

16 Object-oriented design of preconditioned iterative methods in diffpack
Are Magnus Bruaset, Hans Petter Langtangen
March 1997 **ACM Transactions on Mathematical Software (TOMS)**, Volume 23 Issue 1
Publisher: ACM Press
Full text available: pdf(734.26 KB) Additional Information: full citation, references, citings, index terms, review

17 A theory of using history for equational systems with applications
Rakesh M. Verma
September 1995 **Journal of the ACM (JACM)**, Volume 42 Issue 5
Publisher: ACM Press
Full text available: pdf(2.70 MB) Additional Information: full citation, abstract, references, index terms, review

18 High performance Fortran language specification
CORPORATE Rice University
December 1993 **ACM SIGPLAN Fortran Forum**, Volume 12 Issue 4
Publisher: ACM Press
Full text available: pdf(5.69 MB) Additional Information: full citation, abstract, citings, index terms

19 Fortran 8X draft
Loren P. Meissner
December 1989 **ACM SIGPLAN Fortran Forum**, Volume 8 Issue 4
Publisher: ACM Press
Full text available: pdf(21.36 MB) Additional Information: full citation, abstract, index terms

20 The design of the E programming language
Joel E. Richardson, Michael J. Carey, Daniel T. Schuh
July 1993 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 15 Issue 3
Publisher: ACM Press
Full text available: pdf(2.78 MB) Additional Information: full citation, references, citings, index terms, review

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Terms used **anonymous class java** Found **8,797** of **167,655**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 Chianti: a tool for change impact analysis of java programs
Xiaoxia Ren, Fenil Shah, Frank Tip, Barbara G. Ryder, Ophelia Chesley
October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '04**, Volume 39 Issue 10
Publisher: ACM Press
Full text available: pdf(465.37 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper reports on the design and implementation of Chianti, a change impact analysis tool for Java that is implemented in the context of the Eclipse environment. Chianti analyzes two versions of an application and decomposes their difference into a set of atomic changes. Change impact is then reported in terms of affected (regression or unit) tests whose execution behavior may have been modified by the applied changes. For each affected test, Chianti also determines a set of affecting cha ...

Keywords: analysis of object-oriented programs, change impact analysis, regression test, unit test

2 On objects and events
Patrick Th. Eugster, Rachid Guerraoui, Christian Heide Damm
October 2001 **ACM SIGPLAN Notices , Proceedings of the 16th ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications OOPSLA '01**, Volume 36 Issue 11
Publisher: ACM Press
Full text available: pdf(308.58 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents linguistic primitives for publish/subscribe programming using events and objects. We integrate our primitives into a strongly typed object-oriented language through four mechnisms: (1) serialization, (2) multiple subtyping, (3)closures, and (4) deferred code evaluation. We illustrate our primitives through Java, showing how we have overcome its respective lacks. A precompiler transforms statements based on our publish/subscribe primitives into calls to specifically generated ...

3 Scalable extensibility via nested inheritance
Nathaniel Nystrom, Stephen Chong, Andrew C. Myers
October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN**

conference on Object-oriented programming, systems, languages, and applications OOPSLA '04, Volume 39 Issue 10

Publisher: ACM Press

Full text available: pdf(196.74 KB) Additional Information: full citation, abstract, references, index terms

Inheritance is a useful mechanism for factoring and reusing code. However, it has limitations for building extensible systems. We describe nested inheritance, a mechanism that addresses some of the limitations of ordinary inheritance and other code reuse mechanisms. Using our experience with an extensible compiler framework, we show how nested inheritance can be used to construct highly extensible software frameworks. The essential aspects of nested inheritance are formalized i ...

Keywords: inheritance, nested classes, object-oriented programming languages, virtual classes

4 Extensible algebraic datatypes with defaults

Matthias Zenger, Martin Odersky

October 2001 **ACM SIGPLAN Notices , Proceedings of the sixth ACM SIGPLAN international conference on Functional programming ICFP '01**, Volume 36 Issue 10

Publisher: ACM Press

Full text available: pdf(259.40 KB) Additional Information: full citation, abstract, references, citings, index terms

A major problem for writing extensible software arises when recursively defined datatypes and operations on these types have to be extended simultaneously without modifying existing code. This paper introduces Extensible Algebraic Datatypes with defaults, which promote a simple programming pattern to solve this well-known problem. We show that it is possible to encode extensible algebraic datatypes in an object-oriented language, using a new design pattern for extensible visitors. Extensible alg ...

5 Technical correspondence: Closures for statically-typed object-oriented languages

José de Oliveira Guimarães

August 2004 **ACM SIGPLAN Notices**, Volume 39 Issue 8

Publisher: ACM Press

Full text available: pdf(117.69 KB) Additional Information: full citation, abstract, references

Closures increase considerably the level of a language by mixing access to local variables with remote execution of a set of locally-defined statements. However, to date closures have not been added to statically-typed languages because it is difficult to type them and runtime errors occur if local variables that no longer exist are accessed. We proposed a limited but quite general kind of closure for statically-typed object-oriented languages. They can be used in most situations normal closu ...

Keywords: Smalltalk blocks, closure, green, object-oriented languages

6 Converting java programs to use generic libraries

Alan Donovan, Adam Kiežun, Matthew S. Tschantz, Michael D. Ernst

October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '04**, Volume 39 Issue 10

Publisher: ACM Press

Full text available: pdf(1.18 MB) Additional Information: full citation, abstract, references, index terms

Java 1.5 will include a type system (called JSR-14) that supports parametric polymorphism, or generic classes. This will bring many benefits to Java

programmers, not least because current Java practice makes heavy use of logically-generic classes, including container classes.

Translation of Java source code into semantically equivalent JSR-14 source code requires two steps: parameterization (adding type parameters to class definitions) and instantiation (a ...

Keywords: JSR-14, Java 1.5, Java 5, generic types, instantiation types, parameterized types, parametric polymorphism, raw types, type inference

7 Making the future safe for the past: adding genericity to the Java programming language

Gilad Bracha, Martin Odersky, David Stoutamire, Philip Wadler

October 1998 **ACM SIGPLAN Notices , Proceedings of the 13th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '98**, Volume 33 Issue 10

Publisher: ACM Press

Full text available: pdf(1.91 MB) Additional Information: full citation, abstract, references, citings, index terms

We present GJ, a design that extends the Java programming language with generic types and methods. These are both explained and implemented by translation into the unextended language. The translation closely mimics the way generics are emulated by programmers: it erases all type parameters, maps type variables to their bounds, and inserts casts where needed. Some subtleties of the translation are caused by the handling of overriding.GJ increases expressiveness and safety: code utilizing generic ...

8 Technical papers: software maintenance: Concern graphs: finding and describing concerns using structural program dependencies

Martin P. Robillard, Gail C. Murphy

May 2002 **Proceedings of the 24th International Conference on Software Engineering**

Publisher: ACM Press

Full text available: pdf(1.35 MB) Additional Information: full citation, abstract, references, citings, index terms

Many maintenance tasks address concerns, or features, that are not well modularized in the source code comprising a system. Existing approaches available to help software developers locate and manage scattered concerns use a representation based on lines of source code, complicating the analysis of the concerns. In this paper, we introduce the Concern Graph representation that abstracts the implementation details of a concern and makes explicit the relationships between different parts of the co ...

9 Performance and Analysis: WebGraph: things you thought you could not do with Java™

Paolo Boldi, Sebastiano Vigna

June 2004 **Proceedings of the 3rd international symposium on Principles and practice of programming in Java PPPJ '04**

Publisher: Trinity College Dublin

Full text available: pdf(180.33 KB) Additional Information: full citation, abstract, references

Studying web graphs is often difficult due to their large size. The WebGraph framework is a suite of codes, algorithms and tools that make it easy to manipulate large web graphs, and to store them in a limited space, by exploiting the inner redundancies of the web. WebGraph is based on sophisticated bitwise compression techniques, and functional-style lazy constructions. Common wisdom would say that the most unlikely language to

implement such a framework is Java. We are going to tell you the re ...

10 Compatible genericity with run-time types for the Java programming language

Robert Cartwright, Guy L. Steele

October 1998 **ACM SIGPLAN Notices , Proceedings of the 13th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '98**, Volume 33 Issue 10

Publisher: ACM Press

Full text available: pdf(1.97 MB) Additional Information: full citation, abstract, references, citings, index terms

The most serious impediment to writing substantial programs in the Java™ programming language is the lack of a *gentricity* mechanism for abstracting classes and methods with respect to type. During the past two years, several research groups have developed Java extensions that support various forms of genericity, but none has succeeded in accommodating general type parameterization (akin to Java arrays) while retaining compatibility with the existing. Java Virtual Machine. In thi ...

11 Language-specific make technology for the Java programming language

Mikhail Dmitriev

November 2002 **ACM SIGPLAN Notices , Proceedings of the 17th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '02**, Volume 37 Issue 11

Publisher: ACM Press

Full text available: pdf(238.19 KB) Additional Information: full citation, abstract, references, citings, index terms

Keeping the code of a Java application consistent (code is consistent if all of the project classes can be recompiled together without errors) prevents late linking errors, and thus may significantly improve development turnaround time. In this paper we describe a make technology for the Java programming language, that is based on smart dependency checking, guarantees consistency of the project code, and at the same time reduces the number of source code recompilations to the minimum. After proj ...

Keywords: Java, dependency checking, development turnaround time, make

12 A practical type system and language for reference immutability

Adrian Birka, Michael D. Ernst

October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '04**, Volume 39 Issue 10

Publisher: ACM Press

Full text available: pdf(171.73 KB) Additional Information: full citation, abstract, references, index terms

This paper describes a type system that is capable of expressing and enforcing immutability constraints. The specific constraint expressed is that the abstract state of the object to which an immutable reference refers cannot be modified using that reference. The abstract state is (part of) the transitively reachable state: that is, the state of the object and all state reachable from it by following references. The type system permits explicitly excluding fields or objects from the abstract ...

Keywords: Java, Javari, const, immutability, mutable, readonly, type system, verification

13 Type types: Scalable component abstractions

Martin Odersky, Matthias Zenger
October 2005 **Proceedings of the 20th annual ACM SIGPLAN conference on Object oriented programming systems languages and applications OOPSLA '05**
Publisher: ACM Press
Full text available: pdf(650.88 KB) Additional Information: full citation, abstract, references, index terms

We identify three programming language abstractions for the construction of reusable components: abstract type members, explicit selftypes, and modular mixin composition. Together, these abstractions enable us to transform an arbitrary assembly of static program parts with hard references between them into a system of reusable components. The transformation maintains the structure of the original system. We demonstrate this approach in two case studies, a subject/observer framework and a compile ...

Keywords: Scala, abstract types, classes, components, mixins

14 A selective, just-in-time aspect weaver
Yoshiki Sato, Shigeru Chiba, Michiaki Tatsubori
September 2003 **Proceedings of the second international conference on Generative programming and component engineering GPCE '03**
Publisher: Springer-Verlag New York, Inc.
Full text available: pdf(256.62 KB) Additional Information: full citation, abstract, references, citings, index terms

Dynamic AOP (Aspect-Oriented Programming) is receiving growing interests in both the academia and the industry. Since it allows weaving aspects with a program at runtime, it is useful for rapid prototyping and adaptive software. However, the previous implementations of dynamic AOP systems suffered from serious performance penalties. This paper presents our new efficient dynamic AOP system in Java for addressing the underlying problem. This system called Wool is a hybrid of two approaches. When a ...

15 Optimising aspectJ
Pavel Avgustinov, Aske Simon Christensen, Laurie Hendren, Sascha Kuzins, Jennifer Lhoták, Ondřej Lhoták, Oege de Moor, Damien Sereni, Ganesh Sittampalam, Julian Tibble
June 2005 **ACM SIGPLAN Notices , Proceedings of the 2005 ACM SIGPLAN conference on Programming language design and implementation PLDI '05**, Volume 40 Issue 6
Publisher: ACM Press
Full text available: pdf(194.20 KB) Additional Information: full citation, abstract, references, index terms

AspectJ, an aspect-oriented extension of Java, is becoming increasingly popular. However, not much work has been directed at optimising compilers for AspectJ. Optimising AOP languages provides many new and interesting challenges for compiler writers, and this paper identifies and addresses three such challenges.First, compiling *around* advice efficiently is particularly challenging. We provide a new code generation strategy for *around* advice, which (unlike previous implementations) ...

Keywords: around advice, aspect-oriented programming language, aspectJ, cflow pointcut, optimization

16 Semantic interfaces and OWL tools: Parsing owl dl: trees or triples?
Sean K. Bechhofer, Jeremy J. Carroll
May 2004 **Proceedings of the 13th international conference on World Wide Web**
Publisher: ACM Press
Full text available: pdf(156.25 KB) Additional Information: full citation, abstract, references, citings, index terms

The Web Ontology Language (OWL) defines three classes of documents: Lite, DL, and Full. All RDF/XML documents are OWL Full documents, some OWL Full documents are also OWL DL documents, and some OWL DL documents are also OWL Lite documents. This paper discusses *parsing* and *species recognition* -- that is the process of determining whether a given document falls into the OWL Lite, DL or Full class. Wedescribe two alternative approaches to this task, one based on abstract syntax trees, ...

Keywords: owl, parsing, rdf, semantic web

17 Advanced control flow in Java card programming

Peng Li, Steve Zdancewic

June 2004 **ACM SIGPLAN Notices , Proceedings of the 2004 ACM SIGPLAN/SIGBED conference on Languages, compilers, and tools for embedded systems LCTES '04**, Volume 39 Issue 7

Publisher: ACM Press

Full text available: pdf(205.46 KB) Additional Information: full citation, abstract, references, index terms

Java Card technology simplifies the development of smart card applications by providing a high-level programming language similar to Java. However, the master-slave programming model used in current Java Card platform creates control flow difficulties when writing complex card programs, making it inconvenient, tedious, and error-prone to implement Java Card applications. This paper examines these drawbacks of the master-slave model and proposes a concurrent thread model for developing future Jav ...

Keywords: CPS, Java card, continuation, control flow, smart card, trampolined style

18 Jedd: a BDD-based relational extension of Java

Ondřej Lhoták, Laurie Hendren

June 2004 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2004 conference on Programming language design and implementation PLDI '04**, Volume 39 Issue 6

Publisher: ACM Press

Full text available: pdf(137.26 KB) Additional Information: full citation, abstract, references, citings, index terms

In this paper we present Jedd, a language extension to Java that supports a convenient way of programming with Binary Decision Diagrams (BDDs). The Jedd language abstracts BDDs as database-style relations and operations on relations, and provides static type rules to ensure that relational operations are used correctly.The paper provides a description of the Jedd language and reports on the design and implementation of the Jedd translator and associated runtime system. Of particular interest is ...

Keywords: Java, binary decision diagrams, boolean formula satisfiability, language design, program analysis, relations

19 Profiling Java applications using code hotswapping and dynamic call graph revelation

Mikhail Dmitriev

January 2004 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 4th international workshop on Software and performance WOSP '04**, Volume 29 Issue 1

Publisher: ACM Press

Full text available: pdf(1.32 MB) Additional Information: full citation, abstract, references

Instrumentation-based profiling has many advantages and one serious disadvantage: usually high performance overhead. This overhead can be substantially reduced if only a

small part of the target application (for example, one that has previously been identified as a performance bottleneck) is instrumented, while the rest of the application code continues to run at full speed. The value of such a profiling technology would increase further if the code could be instrumented and de-instrumented as m ...

20 A practical comparison between Java and Ada in implementing a real-time embedded system

Eric Potratz

December 2003 **ACM SIGAda Ada Letters , Proceedings of the 2003 annual ACM SIGAda international conference on Ada: the engineering of correct and reliable software for real-time & distributed systems using ada and related technologies SigAda '03**, Volume XXIV Issue 1

Publisher: ACM Press

Full text available: pdf(259.58 KB) Additional Information: full citation, abstract, references, citings, index terms, review

This paper presents a student's observations from an undergraduate research project that explored using Java to implement the software for a real-time embedded system that was originally implemented in a university-level real-time systems course using Ada 95. It briefly gives an overview of the project, the decision made concerning which Java virtual machine to use, and how that virtual machine performed in the real-time environment. It then goes into detail about the merits and drawbacks of usi ...

Keywords: Ada, Java, concurrency, conditional synchronization, drivers, embedded systems, memory management, object-oriented programming, package elaboration, performance, priority inversion, real-time systems, scheduling

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Results for "(template class<in>metadata)" e-mail
Your search matched **8** of **1286976** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

» Search Options

View Session History

New Search

Modify Search

(template class<in>metadata)

Check to search only within this results set

Display Format: Citation Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

Select Article Information

1. **A parameterized cost model to order classes for class-based testing of C**
Malloy, B.A.; Clarke, P.J.; Lloyd, E.L.;
Software Reliability Engineering, 2003. ISSRE 2003. 14th International Sympo
17-20 Nov. 2003 Page(s):353 - 364
Digital Object Identifier 10.1109/ISSRE.2003.1251057
AbstractPlus | Full Text: PDF(632 KB) IEEE CNF

2. **Statistical error modeling of CNN-UM architectures: the binary case**
Foldesy, P.;
Cellular Neural Networks and Their Applications, 2002. (CNNA 2002). Proceec
7th IEEE International Workshop on
22-24 July 2002 Page(s):467 - 474
Digital Object Identifier 10.1109/CNNA.2002.1035085
AbstractPlus | Full Text: PDF(301 KB) IEEE CNF

3. **LOCAL++: a C++ framework for local search algorithms**
Schaerf, A.; Lenzerini, M.; Cadoli, M.;
Technology of Object-Oriented Languages and Systems, 1999. Proceedings o
7-10 June 1999 Page(s):152 - 161
Digital Object Identifier 10.1109/TOOLS.1999.779008
AbstractPlus | Full Text: PDF(264 KB) IEEE CNF

4. **Intelligent agent platform for procurement**
Subramanian, K.; Lee, S.; Tey Kar Shiang; Gan Beng Sue;
Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceeding
International Conference on
Volume 3, 12-15 Oct. 1999 Page(s):107 - 112 vol.3
Digital Object Identifier 10.1109/ICSMC.1999.823163
AbstractPlus | Full Text: PDF(596 KB) IEEE CNF

5. **An analysis of CNN settling time**
Hanggi, M.; Moschytz, G.S.;
Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE Interna
on
Volume 3, 31 May-3 June 1998 Page(s):155 - 158 vol.3
Digital Object Identifier 10.1109/ISCAS.1998.703934
AbstractPlus | Full Text: PDF(312 KB) IEEE CNF

6. **New spatial-temporal patterns and the first programmable on-chip bifurc**
Petras, I.; Roska, T.; Chua, L.O.;
Circuits and Systems I: Fundamental Theory and Applications, IEEE Transacti
Circuits and Systems I: Regular Papers, IEEE Transactions on]
Volume 50, Issue 5, May 2003 Page(s):619 - 633
Digital Object Identifier 10.1109/TCSI.2003.811022
AbstractPlus | References | Full Text: PDF(2657 KB) IEEE JNL

7. **Developing a user-level middleware for out-of-core computation on Grids**
Jianqi Tang; Binxing Fang; Mingzeng Hu; Hongli Zhang;
Cluster Computing and the Grid, 2004. CCGrid 2004. IEEE International Symp
19-22 April 2004 Page(s):686 - 690
Digital Object Identifier 10.1109/CCGrid.2004.1336699
AbstractPlus | Full Text: PDF(384 KB) IEEE CNF

8. **New spatial-temporal patterns and the first programmable on-chip bifurc**
Petras, I.; Roska, T.; Chua, L.O.;
Cellular Neural Networks and Their Applications, 2002. (CNNA 2002). Proceec
7th IEEE International Workshop on
22-24 July 2002 Page(s):25 - 32
Digital Object Identifier 10.1109/CNNA.2002.1035031
AbstractPlus | Full Text: PDF(418 KB) IEEE CNF